

(19) 日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平 10 - 105499

(43) 公開日 平成10年(1998)4月24日

(51) Int. Cl. °	識別記号	F I
G 0 6 F 13/00	3 5 7	G 0 6 F 13/00 3 5 7 Z
	3 5 1	3 5 1 H
3/14	3 1 0	3/14 3 1 0 A
9/46	3 6 0	9/46 3 6 0 B
15/00	3 1 0	15/00 3 1 0 R
審査請求 未請求 請求項の数 5		OL (全 8 頁)

(21) 出願番号 特願平 9 - 1 6 8 7 8 7

(22) 出願日 平成 9 年 (1997) 6 月 25 日

(31) 優先権主張番号 特願平 8 - 2 0 3 5 7 4

(32) 優先日 平 8 (1 9 9 6) 8 月 1 日

(33) 優先権主張国 日本 (J P)

(71) 出願人 000005223
富士通株式会社
神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

(72) 発明者 原 裕貴
神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

(72) 発明者 金谷 延幸
神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号 富士通株式会社内

(74) 代理人 弁理士 長澤 俊一郎 (外 1 名)

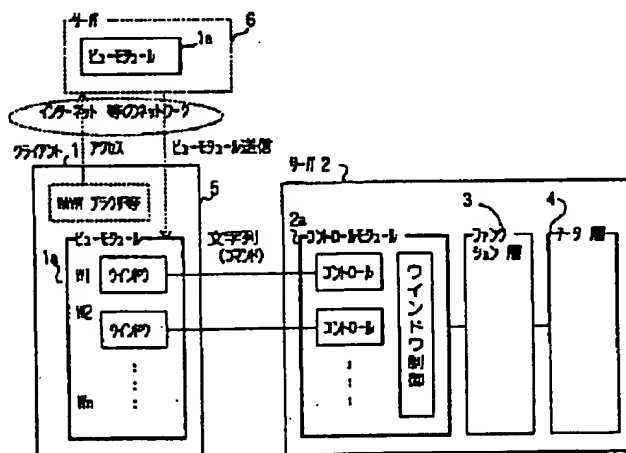
(54) 【発明の名称】 クライアント／サーバシステムおよびクライアント／サーバシステムに使用されるプログラムを記録した記録媒体

(57) 【要約】

【課題】 クライアント側に置かれたモジュールを完全にアプリケーションに対して独立し、また、クライアント・サーバ間の通信量を少なくすること。

【解決手段】 3 階層システムにおけるプレゼンテーション層をビューモジュール 1 a とコントロールモジュール 2 a に分離する。クライアント 1 のビューモジュール 1 a は画面上の各ウィンドウ W1 ~ Wn に対応しており、画面上の操作に従ってコマンドや文字列をサーバ 2 のコントロールモジュール 2 a に渡す。コントロールモジュール 2 a は、各ウィンドウ W1 ~ Wn から送られてきたコマンドや文字列を解釈して、必要に応じてファンクション層 3 のサービスを起動する。また、コントロールモジュール 2 a はウィンドウ W1 ~ Wn の遷移の管理等を行う。なお、WWW ブラウザ等を利用してインターネット等を介してサーバ 6 等からクライアント 1 側にビューモジュール 1 a を転送することもできる。

本発明の原理図



【特許請求の範囲】

【請求項1】 ウィンドウ等のグラフィック・インタフェースもしくはテキスト・インタフェースを持つクライアント・マシンと、データとそれに対するサービス機能を備えたサーバマシンから構成されるクライアント/サーバシステムにおいて、

各ウィンドウの制御を行うビューモジュールと、ウィンドウ間の制御とアプリケーションモジュールが置かれたファンクション層との通信を行うコントロールモジュールとを分離して設けたことを特徴とするクライアント/サーバシステム。

【請求項2】 ビューモジュールをクライアント側に配置し、コントロールモジュールをサーバ側に配置したことを特徴とする請求項1のクライアント/サーバシステム。

【請求項3】 クライアントにモジュールの転送を依頼する手段を設け、該手段を利用して、実行時、ネットワークを介してビューモジュールをサーバからクライアントに転送することを特徴とする請求項1または請求項2のクライアント/サーバシステム。

【請求項4】 ウィンドウ等のグラフィック・インタフェースもしくはテキスト・インタフェースを持つクライアント・マシンと、データとそれに対するサービス機能を備えたサーバマシンから構成されるクライアント/サーバシステムに使用されるプログラムを記録した記録媒体であって、

上記プログラムは、各ウィンドウの制御を行うビューモジュールと、ウィンドウ間の制御とアプリケーションモジュールが置かれたファンクション層との通信を行うコントロールモジュールとが分離されて構成されることを特徴とするプログラムを記録した記録媒体。

【請求項5】 ビューモジュールがクライアント側に配置されるプログラムであり、コントロールモジュールがサーバ側に配置されるプログラムであることを特徴とする請求項4のプログラムを記録した記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、ウィンドウ等のグラフィック・インタフェースを持つクライアントマシンと、RDB（リレーショナル・データベース）等の情報と、それに対するサービスを持つサーバマシンから構成されるクライアント/サーバシステムおよび該クライアント/サーバシステムに使用されるプログラムを記録した記録媒体に関し、特に本発明は、クライアント/サーバ間の通信量を少なくでき、また、クライアント側のモジュールをアプリケーションから完全に独立にすることができるクライアント/サーバシステムおよび該クライアント/サーバシステムに使用されるプログラムを記録した記録媒体に関するものである。

【0002】

【従来の技術】ネットワーク環境におけるコンピュータシステムは、従来の中央集中型のホストと端末という形態から、クライアント/サーバ型に変化した。しかし、分散システムの大規模化にともない、クライアントの負荷が大きい、クライアント/サーバ間の通信量が多い等の問題点が生じてきている。そこで、これらの問題に対処するため、プレゼンテーション層/ファンクション層/データ層からなる3層アーキテクチャが提案され、実際にシステムも構築されている。

【0003】図8は上記した3階層システムの構成を示す図である。同図において、50はクライアント、60はサーバである。クライアント50には、プレゼンテーション層51が設けられており、プレゼンテーション層51は各ウィンドウWを制御したり、各ウィンドウW間の遷移を制御するウィンドウ制御部51aと、サーバ60のファンクション層61との通信を行うインタフェース部51bから構成されている。サーバ60には、ファンクション層61とデータ層62が設けられており、ファンクション層61は、上記プレゼンテーション層51と通信を行うインタフェース部61aと各種アプリケーションプログラムを実行するアプリケーション・モジュール61bから構成されている。また、データ層62は、データ62bと、該データ62bにアクセスを行うためのデータアクセス・モジュール62aから構成されている。

【0004】上記した3階層システムは、GUI（グラフィカル・ユーザ・インタフェース）モジュールと業務ロジックモジュール（アプリケーション・モジュール）が分かれるので、従来のアプリケーション・モジュールをクライアント側に配置した2階層システムと比べ、クライアント/サーバ間の通信量が少なく、開発が容易であり、システムの変更にも容易に対応できるといった特徴を持つ。

【0005】しかしながら、上記3階層アーキテクチャでも、クライアント50側に置かれるプレゼンテーション層51はサーバ60側に置かれるファンクション層61と通信を行うので、業務内容に依存せざるを得ず、システム変更やインターネットでの運用において問題が生ずる。すなわち、プレゼンテーション層51は通常クライアント50側に配置され、このモジュールがファンクション層61に出す内容は、どうしてもアプリケーションに依存する。このため、アプリケーションが変更された場合、それに応じてクライアント側のGUIも変更せざるを得ず、システムの変更に対して影響を受けやすい。

【0006】また、インターネットを用いてクライアント/サーバシステムを構築することを考えた場合、近年J A V A等を用いてサーバからプログラムを送ってクライアント上で実行する方式が注目されているが、インタ

一ネットを介して送られた汎用的なプログラムを実行する場合、アプリケーションに依存するモジュールをクライアント側におくのは、システムの維持などの点から好ましくない。

【0007】一方、近年、全くアプリケーションに依存しないウィンドウ・システムとして、Xウィンドウ・システムが提案されている。図9は上記したXウィンドウ・システムの構成を示す図である。同図において、70はXサーバ、80はクライアントである（Xウィンドウ・システムでは上記図8のクライアント50、サーバ60に相当するものがそれぞれXサーバ70、クライアント80である）。Xウィンドウ・システムにおいて、Xサーバ70はウィンドウを生成し表示画面上に表示させたり、キーボード、マウス入力処理する。また、クライアント80側にはアプリケーション・モジュールが置かれており、クライアント80はXサーバ70に対してウィンドウの生成を要求したり、図形出力の要求をする。

【0008】これらの要求は、クライアント・サーバ間の基本メッセージであるXプロトコルによりXサーバ70に転送される。上記Xプロトコルには、クライアント80からXサーバ70に転送される「リクエスト」、Xサーバ70からクライアント80に転送される「リプライ」、「イベント」、「エラー」の4種類がある。クライアント80のアプリケーションは、直接上記Xプロトコルをハンドリングするのではなく、Klibというライブラリの関数をコールし、コールされたKlibがXプロトコルを発行する。

【0009】図10は上記Xウィンドウ・システムにおけるXサーバ70とクライアント80間における通信を説明する図であり、同図に示すように、Xサーバ70から、マウスクリック、文字入力等のイベントが入力されると、クライアント80はこれらのイベントにตอบสนองして、新規ウィンドウオープン、文字表示等のリクエストをXサーバ70に返し、Xサーバ70はこれに対応した所定の処理を行う。

【0010】上記したXウィンドウ・システムにおいては、表示画面にウィンドウを表示したりキーボード入力等のデバイスに依存する操作はXサーバ70が担当するため、クライアント側のアプリケーションはデバイスに依存することがない。また、Xサーバ70側のOSやハードウェアに依存した動作はXサーバ70が処理を担当するため、クライアント80は一般に相手システムがXプロトコルを理解できさえすれば、OSやベンダに依存せずにプログラムを書くことができるといった特徴を持っている。

【0011】しかしながら、Xウィンドウ・システムが提供するイベントは図10に示したように、マウスクリックとか、文字入力等の非常に低いレベルの情報なので、クライアント80とXサーバ70間に多くの通信が

生ずる。また、Xサーバ側ではこれらのイベント列を解釈する必要があるので、アーキテクチャとして複雑となり、GUIが変更されると多くの変更を行う必要が生ずる。

【0012】

【発明が解決しようとする課題】以上のように、図8に示した3階層システムは、クライアント側に置かれたモジュールがアプリケーションに依存するため、システム変更やインターネットでの運用において問題点があった。また、図9に示したXウィンドウ・システムは、クライアント80とXサーバ70間に多くの通信が生じ、また、アーキテクチャとして複雑でありGUIの変更に伴い、多くの変更が生ずるといった問題点があった。

【0013】本発明は上記した事情に鑑みなされてものであって、その目的とするところは、クライアント側に置かれたモジュールを完全にアプリケーションに対して独立させ、また、クライアント・サーバ間の通信量を少なくすることである。

【0014】

【課題を解決するための手段】図1は本発明の原理図であり、本発明においては同図に示すように従来の3階層システムにおけるプレゼンテーション層を、ビューモジュールとコントロールモジュールに分離する。同図において、1はクライアント、1aはビューモジュール、W1～Wnはウィンドウ（ビュー）、2はサーバ、2aはコントロールモジュール、3はファンクション層、4はデータ層である。

【0015】同図において、ビューモジュール1aは画面上の各ウィンドウW1～Wnに対応しており、画面上の操作に従って、コマンドや文字列をコントロールモジュール2aに渡す。ここで、各ウィンドウW1～Wnはそこでの操作が何を意味しており、どのように使われるのか一切知らなくてよい。このため、ビューモジュール1aはアプリケーションに依存せずに作成することができる。また各ウィンドウW1～Wnは自分自身の動作のみについて知っていればよく、ウィンドウ間の遷移などについては知る必要はない。また、コントロールモジュール2aは、各ウィンドウW1～Wnから送られてきたコマンドや文字列を解釈して、必要に応じてファンクション層3のサービスを起動する。また、コントロールモジュール2aはウィンドウW1～Wnを束ねてその遷移の管理を行う。

【0016】このような構成とすることにより、クライアント1上のビューモジュール1aは完全にアプリケーションから独立になるので、システムの設計が容易となり、また、変更に対しても容易に対応できる。なお、図1においては、コントロールモジュール2aをサーバ2側に置いているが、コントロールモジュール2aをクライアント1側においても、上記と同様な効果が得られる。一方、JAVA等を用いてインターネット上でクラ

クライアント1にプログラムを送る場合にも、クライアント1側はアプリケーションから独立しているため、容易に対応することができる。

【0017】さらに、図1に示すように、WWW (World Wide Web) ブラウザ5のようなインターネット等のネットワークを介してクライアント1側にプログラムを転送する機能を利用することにより、インターネット等を介してサーバ6 (サーバ2と同じサーバでもよい) 等から、クライアント1側にビューモジュール1aを転送することもできる。このようにすることにより、クライアント1側に予めビューモジュール1aが用意されていなくても、実行時、ビューモジュール1aをクライアント1に転送できるので、本発明のシステムの適用範囲を一層拡大することができる。

【0018】本発明は上記点に基づきなされたものであり、次のようにして前記課題を解決する。

(1) ウィンドウ等のグラフィック・インタフェースもしくはテキスト・インタフェースを持つクライアント・マシンと、データとそれに対するサービス機能を備えたサーバマシンから構成されるクライアント/サーバシステムにおいて、各ウィンドウの制御を行うビューモジュールと、ウィンドウ間の制御とアプリケーションモジュールが置かれたファンクション層との通信を行うコントロールモジュールとを分離する。

(2) 上記(1)において、ビューモジュールをクライアント側に配置し、コントロールモジュールをサーバ側に配置する。

(3) 上記(1)(2)において、クライアントにモジュールの転送を依頼する手段を設け、該手段を利用して、実行時、ネットワークを介してビューモジュールをサーバからクライアントに転送する。

【0019】

【発明の実施の形態】図2は本発明の実施例のシステムの構成を示す図である。同図において、10はクライアントであり、クライアント10には、各ウィンドウW1~Wnに対応するビューモジュール11が置かれている。20はサーバであり、サーバ20にはコントロールモジュール21と、ファンクション層22と、データ層23が置かれている。コントロールモジュール21には、ビューモジュール11の各ウィンドウW1~Wnに対応して創成されるコントロール21a、ウィンドウ制御部21bとが設けられ、ウィンドウ制御部21bによりビューモジュール11のウィンドウW1~Wnの動作、およびその遷移が制御される。

【0020】また、コントロールモジュール21に設けられたインタフェース部21cによりファンクション層22との通信が行われる。ファンクション層22には上記インタフェース部21cと通信を行うインタフェース部22aと、各種業務処理を行うアプリケーション・モジュール22bが置かれており、アプリケーション・モ

ジュール22bはデータ層23のデータアクセスモジュール23aを介してデータ23bにアクセスする。

【0021】図3は図2に示したシステムの動作を説明する図であり、同図により図2のシステムの動作を説明する。まず、システムを立ち上げると、クライアント10の画面上には、各種作業を選択するためのトップウィンドウが表示される。作業者がトップウィンドウのメニューから作業を指定すると、作業表示がサーバ20のコントロールモジュール21に送られ、コントロールモジュール21は作業表示指示に応じて作業ウィンドウオープンに応答と、作業内容に応じたビューモジュールをクライアント側に送る。また、それに対応したコントロール21aを創成する。これにより、クライアント10の表示画面上には上記作業ウィンドウが開く。

【0022】これに応じて、作業者は上記作業ウィンドウ上でキーボード、マウス等を使用した文字編集作業を行う。これらの編集処理はクライアント10のビューモジュール11上で行われ、上記操作内容はコントロールモジュール21へは送られない。上記のような編集処理が終了すると、クライアント10のビューモジュール11からサーバ21のコントロールモジュール21に対して編集処理の終了を示す文字列(コマンド)が送られる。ここで、ビューモジュール11は上記編集終了通知を文字列として認識しているだけであり、その操作が何を意味するのか、またどのように使われるのか意識しない。コントロールモジュール21のコントロール21aは上記文字列が通知されると、送られてきた文字列(コマンド)を解釈してそれに対応した処理を行い、必要に応じてファンクション層22のサービスを起動する。

30 【0023】図4、図5は本実施例のシステムにおいてクライアント側の表示装置に表示される画面の一例を示す図、図6はそのときのクライアントおよびサーバの動作を説明する図であり、図4、図5、図6により本発明の実施例のシステムの適用例を説明する。なお、図4、図5における(a)~(d)は図6の(a)~(d)にそれぞれ対応する。

【0024】システムを立ち上げると、まず、クライアント10の表示装置10に図4(a)に示すトップウィンドウが表示され、これに応じて、図6に示すようにサーバ20にはトップウィンドウに対応したコントロールが創成される。トップウィンドウの上部には図4(a)に示すようにメニューが表示され、該メニューの「ウィンドウ」の中から「ログイン」を選択しパスワード等を入力すると、図6に示すようにサーバ20のトップウィンドウのコントロールによりワークリストのビューが生成されるとともに、ワークリストに対応したコントロールが創成される。

【0025】それに応じて、クライアント10の表示装置には、図4(a)に示すように現在作業者に割り当てられている作業内容を示すワークリストが表示される。

ここで、上記ワークリストに表示された作業を実行するため、図4(b)に示すようにプルダウンメニューの「開始」を選択すると、「開始」という文字列がサーバ20に送られる。これに応じて、サーバ20には、作業ウィンドウのコントロールが創成され、また、サーバ20の作業ウィンドウのコントロールが作業ウィンドウのビューをクライアントに転送する(図6)。これに応じて、クライアント10の表示装置には図5(c)に示すように、作業ウィンドウが表示される。

【0026】作業者は、上記作業ウィンドウ上の必要な項目に入力し編集作業等を行う。この処理はクライアント10の作業ウィンドウ上で行われ、この間クライアント、サーバ間で通信は行われない。作業ウィンドウ上での作業が終了し、図5(d)に示すように作業者がプルダウンメニューの「終了」を選択すると、図6に示すように作業ウィンドウ上の編集結果がサーバ20に転送される。これに応じて、サーバ20の作業ウィンドウのコントロールは編集結果を処理し、必要に応じてファンクション層のサービスを起動する。また、作業ウィンドウのコントロールがクライアント10の作業ウィンドウを閉じ、ついで、サーバ20のトップウィンドウのコントロールが作業ウィンドウのコントロールを消去する。

【0027】ここで、クライアント20の各ウィンドウは、各ウィンドウ内における表示の制御を行うが、前記したように各ウィンドウで行われる操作が何を意味し、サーバに転送される文字列(コマンド)が何を意味しているか知る必要はない。またウィンドウ間の遷移等についても知る必要はない。一方、サーバ20の各コントロールは、クライアント10から転送された文字列を解釈実行するだけであり、クライアント10の各ウィンドウにおける表示内容を知る必要はない。

【0028】以上のように、本実施例においては、ビューモジュール11とコントロールモジュール21をそれぞれクライアント側とサーバ側に設け、ビューモジュールにより各ウィンドウの制御を行い、コントロールモジュールによりウィンドウ間の制御とファンクション層との通信を行うようにしたので、クライアント上のビューモジュールをアプリケーション独立とすることができる。また、各ウィンドウ上での作業を各ウィンドウ内で処理し、ウィンドウを閉じるときその作業内容をサーバ側に転送するようにしたので、クライアントとサーバ間の通信量を少なくすることができる。

【0029】図7は本発明の他の実施例を示す図であり、同図はインターネットを介してクライアント側にビューモジュールを転送する実施例を示している。同図において、10はクライアントであり、クライアント10はWWWブラウザのようなインターネットを介してプログラムの転送を依頼する手段を備えている。20はサーバであり、サーバ20は前記したように、コントロール層、ファンクション層、データ層を備えている。30は

WWWサーバであり、WWWサーバにはトップウィンドウのビューモジュール1aが置かれている。

【0030】同図において、クライアント10の表示画面上にウィンドウを開くには、まずWWWブラウザにより、WWWサーバ30にアクセスし(同図①)、WWWサーバからクライアント10にJAVAアプレット(上記ビューモジュール)を送信する(同図②)。クライアント10に上記ビューモジュール(トップウィンドウに対応するビューモジュール)が送られると、送信されたビューモジュールにより、クライアント10とサーバ20との通信路が確立するとともに、クライアント10の表示画面上にトップウィンドウが表示される。

【0031】作業者が上記トップウィンドウの作業メニューを選択すると、サーバ20のコントロールモジュール21と接続が行われ、前記したようにクライアント10の表示画面上に作業ウィンドウが開くとともに、コントロールモジュール21には対応したコントロール21aが創成される。以下、クライアント10とサーバ20間で前記したような手順で処理が行われ所望の作業が実行される。上記構成とすることにより、クライアント10側にトップビューが置かれていなくても、クライアント10がインターネット等のネットワークを介してプログラムの転送を依頼する機能(例えばWWW2ブラウザ等)を備えていれば、本発明のシステムを利用することができ、本システムの適用範囲を拡大することができる。

【0032】

【発明の効果】以上説明したように、本発明においては、以下の効果を得ることができる。

(1)従来の3階層システムにおけるプレゼンテーション層を、ビューモジュールとコントロールモジュールに分離し、ビューモジュールにより各ウィンドウの制御を行い、コントロールモジュールによりウィンドウ間の制御とファンクション層との通信を行うようにしたので、クライアント上のビューモジュールをアプリケーション独立とすることができる。また、各ウィンドウ上での作業を各ウィンドウ内で処理しているので、クライアントとサーバ間の通信量を少なくすることができる。このため、システムの設計が容易となり、また、変更に対しても容易に対応できる。また、インターネット等のネットワークを介してクライアントにプログラムを送る場合にも、クライアント側はアプリケーションから独立しているため、容易に対応することができる。

【0033】(2)インターネット等のネットワークを介してクライアント側にモジュールを転送する機能を利用することにより、インターネット等を介してサーバからクライアント側にビューモジュールを転送することができる。このため、クライアント側に予めビューモジュールが用意されていなくても、本発明のシステムを利用することができ、本発明のシステムの適用範囲を一層広

大することができる。

【図面の簡単な説明】

【図1】本発明の原理図である。

【図2】本発明の実施例のシステムの構成を示す図である。

【図3】図2のシステムの動作を説明する図である。

【図4】本発明のシステムの適用例の表示画面の一例を示す図である。

【図5】本発明のシステムの適用例の表示画面の一例を示す図（続き）である。

【図6】図4、図5の動作を説明する図である。

【図7】ネットワークを介してビューモジュールを転送する実施例を示す図である。

【図8】従来の3階層システムの構成を示す図である。

【図9】従来のXウィンドウ・システムの構成を示す図である。

【図10】Xウィンドウ・システムの動作を説明する図である。

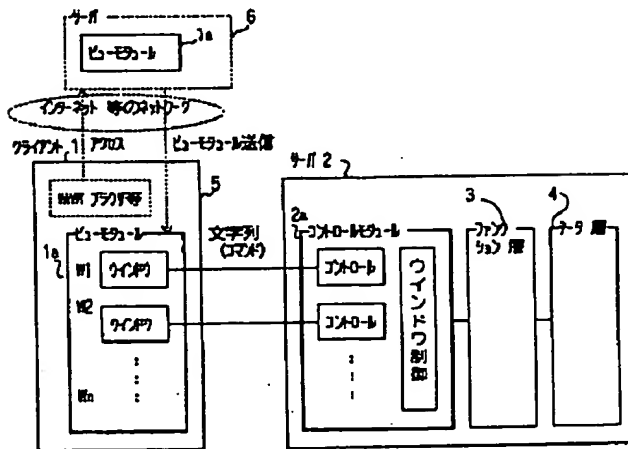
【符号の説明】

1 クライアント
1 a ビューモジュール

2 サーバ
2 a コントロールモジュール
3 ファンクション層
4 データ層
5 プログラムの転送を依頼する手段
6 サーバ
10 クライアント
11 ビューモジュール
20 サーバ
21 コントロールモジュール
21 a コントロール
21 b ウィンドウ制御部
21 c インタフェース部
22 ファンクション層
22 a インタフェース部
22 b アプリケーション・モジュール
23 データ層
23 a データアクセスモジュール
23 b データ
20 30 WWWサーバ
W1~Wn ウィンドウ（ビュー）

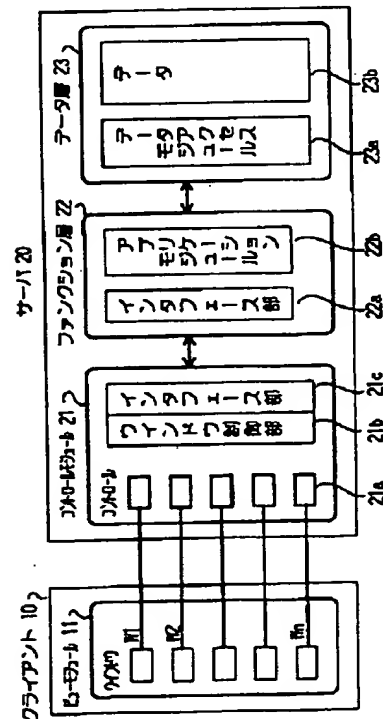
【図1】

本発明の原理図



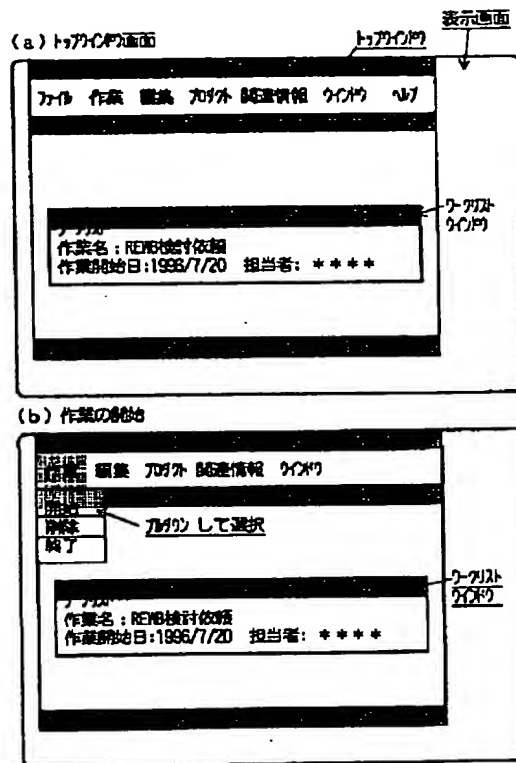
【図2】

本発明の実施例のシステムの構成を示す図



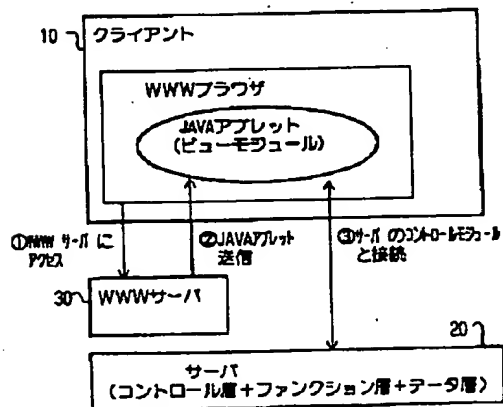
【图 4】

本発明のシステムの適用例の表示画面の一例を示す図



【图 7】

ネットワークを介してビューモジュールを送送する実施例を示す図



【図5】

本発明のシステムの運用例の表示画面の一例を示す図（続き）

(c) 作業開始画面

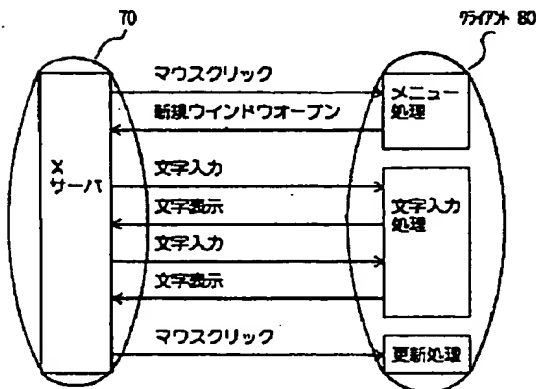
作業 編集 7076 読書情報 9176	
作業名	REMB検討依頼
開始日	1996 7/16
担当者	*****
作業説明	
作業開始	作業終了

(d) 作業終了、7177として選択

作業 編集 7076 読書情報 9176	
開始	終了
作業名	REMB検討依頼
開始日	1996 7/16
担当者	*****
作業説明	
作業開始	作業終了

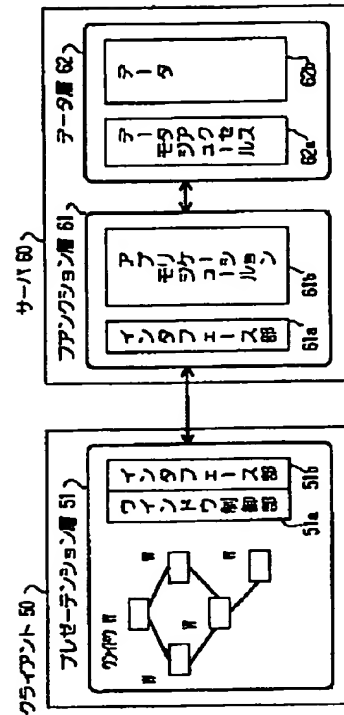
【図10】

Xウィンドウ・システムの動作を説明する図



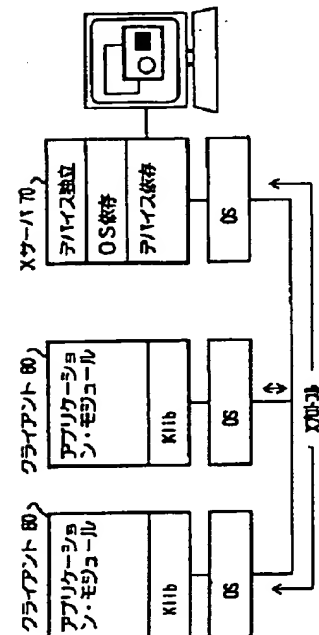
【図8】

従来の3階層システムの構成を示す図



【図9】

従来のXウィンドウ・システムの構成を示す図



PARTIAL TRANSLATION OF JAPANESE UNEXAMINED PATENT PUBLICATION
(KOKAI) NO. 10-105499

[Abstract]

[Object] To make a module placed at a client side fully independent relative to an application and reduce a traffic between a client and a server.

[Means for Attaining the Object] A presentation layer in a three hierarchy layer is divided into a view module 1a and a control module 2a. The view module 1a of a client 1 corresponds to respective windows W1 to Wn on a screen and delivers a command and a character string to the control module 2a of a server in accordance with an operation on the screen. The control module 2a interprets commands and character strings sent from the respective windows W1 to Wn and activates services on a function layer 3 as required. Additionally, the control module 2a manages the transition of the windows W1 to Wn. It is also possible to transfer the view module 1a from a server 6 or the like to the client 1 side via an internet or the like using a WWW browser or the like.

[Claims]

[Claim 1] A client/server system consisting of a client machine having a graphic interface or a text interface such as a window or the like and a server machine comprising data and a service function therefore, characterized in that;

a view module for controlling an individual window and a control module for implementing control between windows and communication with a function layer where an application module is placed are separately provided.

[Claim 2] A client/server system as set forth in Claim 1, wherein said view module is placed on a client side, whereas said control module is placed on a server side.

[Claim 3] A client/server system as set forth in Claim 1 or 2, wherein means for requesting a client to transfer a module

is provided, whereby at the time of execution, said view module is transferred from said server to said client via a network using said means.

[Claim 4] A recording medium having recorded therein a program for use in a client/server system consisting of a client machine having a graphic interface or a text interface such as a window or the like and a server machine comprising data and a service function therefore, characterized in that;

said program consists of a view module for controlling an individual window and a control module for implementing control between windows and communication with a function layer where an application module is placed, said view module and said control module being provided separately.

[Claim 5] A recording medium having recorded therein a program as set forth in Claim 4, wherein said program is a program in which said view module is placed on a client side and a program in which said control module is placed on a server side.

[0012]

[Problems that the Invention is to Solve]

As has been described heretofore, with the three hierarchy layer system shown in Fig. 8, since the module placed on the client side is dependent on the application, there is caused a problem in modification of the system or operation on the internet. In addition, with the X window system shown in Fig. 9, since a lot of communication is produced between the client 80 and the X server 70, and since the system is complicated in terms of architecture, there is caused a problem that a lot of modifications have to be made in conjunction with a modification to the GUI.

[0013]

The present invention was made in view of the situation described above, and an object thereof is to make a module placed on a client side fully independent relative to an

application and reduce the traffic between a client and a server.

[0014]

[Means for Solving the Problems]

Fig. 1 is a diagram showing the principle of the present invention, and in the present invention, as shown in Fig. 1, a presentation layer in a conventional three hierarchy layer system is divided into a view module and a control module. In Fig. 1, reference numeral 1 denotes a client, 1a a view module, W1 to Wn windows (view), 2 a server, 2a a control module, 3 a function layer, and 4 a data layer.

[0015]

In Fig. 1, the view module 1a corresponds to respective windows W1 to Wn on a screen and transfers a command and a character string to the control module 2a in accordance with an operation on the screen. Here, the respective windows W1 to Wn do not have to know at all what is meant by operations thereon or how they are used. Due to this, the view module 1a can be created without depending upon the application. In addition, the respective windows W1 to Wn only have to know about their own operations and do not have to know about transitions between the windows. Additionally, the control module 2a interprets commands and character strings sent from the respective windows W1 to Wn and activates services on the function layer 3. In addition, the control module 2a bundles the windows W1 to Wn and manages transitions thereof.

[0016]

Since the view module 1a on the client 1 becomes fully independent from the application by configuring the presentation layer as described above, the design of the system is eased and a modification can be dealt with easily. In addition, in Fig. 1, while the control module 2a is placed on the server 2 side, even if the control module 2a is placed on the client 1 side, the same advantages as described above

can also be obtained. On the other hand, even in a case where a program is sent to the client 1 on the internet using JAVA or the like, since the client 1 side is independent from the application, this can easily be dealt with.

[0017]

Moreover, as shown in Fig. 1, the view module 1a can be transferred from the server 6 (which may be the same server as the server 2) to the client 1 side via the internet by making use of a function to transfer a program to the client 1 side via a net work such as the internet like the WWW (World wide Web) browser 5. According to this configuration, even if the view module 1a is not prepared on the client 1 side in advance, since the view module 1a can be transferred to the client 1 side at the time of execution, the scope of the application of the present invention can be expanded further.

[0018]

Having been made based on the aforesaid principle thereof, the present invention solves the aforesaid problems as below.

(1) There is provided a client/server system consisting of a client machine having a graphic interface or a text interface such as a window or the like and a server machine comprising data and a service function therefore, characterized in that a view module for controlling an individual window and a control module for implementing control between widows and communication with a function layer where an application module is placed are separately provided.

(2) There is provided a client/server system as set forth under (1) above, wherein the view module is placed on a client side, whereas the control module is placed on a server side.

(3) There is provided a client/server system as set forth under (1) or (2) above, wherein means for requesting a client

to transfer a module is provided, whereby at the time of execution, the view module is transferred from the server to the client via a network using the means.

[0019]

[Mode for Carrying out the Invention]

Fig. 2 is a diagram showing the configuration of a system according to an embodiment of the present invention. In Fig. 2, reference numeral 10 denotes a client, and placed on the client 10 is a view module 11 corresponding to respective windows W1 to Wn. Reference numeral 20 denotes a server, and placed on the server 20 are a control module 21, a function layer 22 and a data layer 23. Provided on the control module 21 are controls 21a that are created so as to correspond to the respective windows W1 to Wn of the view module 1a and a window control unit 21b. The window control unit 21b controls the operations of the windows W1 to Wn of the view module 1a and the transitions thereof.

[0020]

Additionally, an interface unit 21c provided on the control module 21 implements communication with the function layer 22. Placed on the function layer 22 are an interface unit 22a for communicating with the interface unit 21c and an application module 22b for performing various types of work processing, and the application module 22b accesses data 23b via a data access module 23a on the data layer 23.

[0021]

Fig. 3 is a diagram explaining an operation of the system shown in Fig. 2, and an operation of the system will be explained with reference to the same figure. First, when the system is activated, a top window for selecting various types of work is displayed on the screen of the client 10. When the client 10 specifies work from a menu in the top window, a work display is sent to the control module 21 of the server 20, and the control module 21 then sends to the

client side a reply to open a work window in response to the specification of the work display and a view module in response to the contents of the job. In addition, the control module 21 also creates a control 21a corresponding to the view module, whereby the aforesaid work window is opened on the screen of the client 10.

[0022]

In response thereto, the operator performs an editing operation of characters on the work window using a keyboard, a mouse and the like. The editing process is performed on the view module 11 of the client 10, and the operation contents are not sent to the control module 21. When the editing process as described above is completed, a character string (a command) indicating the completion of the editing processing is sent to the control module 21 of the server 21 from the view module 11 of the client 10. Here, the view module only recognizes the editing completion notice as a character string and is not aware of what is meant by the operation or what it is used for. When notified of the character string, the control 21a of the control module 21 interprets a character string (a command) sent thereto, performs a process corresponding to the interpretation, and activates a service on the function layer 22 as required.